AMENDMENTS TO THE SPECIFICATION

IN THE SPECIFICATION:

Please amend the specification as follows. Insertions appear as underlined text (e.g., insertions) while deletions appear as strikethrough text (e.g., ~~strikethrough~~). All previously amended claims appear as clean text.

Please amend the second paragraph beginning on page 22 as follows:

Figure 9B shows the sliding-window transform system of Figure 9A extended to multiple channels. In Figure 9B the output of the A/D 323 is provided to an input of a first sliding-window transform 921, a second sliding-window transform 922, and an M-th sliding window transform ~~922~~ 923 . An output of the first sliding-window transform 921 is provided to an input of a first demapper 931. An output of the second sliding-window transform ~~921~~ 922 is provided to an input of a second demapper 932. An output of the first sliding-window transform 921 is provided to an input of an M-th demapper 933.

Please amend the fifth paragraph beginning on page 24 as follows:

The recursive relation for the above equation is as follows:

$$\xcancel{X_n(\omega_\ell) = X_{n-1}(\omega_\ell) + x[n]e^{-j\omega_\ell n} - x[n-N]e^{-j\omega_\ell (n-N)}}$$

$$\underline{\hat{X}_n(\omega_\ell) = \hat{X}_{n-1}(\omega_\ell) + x[n]e^{-j\omega_\ell n} - x[n-N]e^{-j\omega_\ell (n-N)}}$$

Please amend the last paragraph beginning on page 24 as follows:

Here the new output equals the previous output with the newest mixed input added and the oldest mixed input subtracted. Noting that $e^{-j\omega_\ell n} = e^{-j\omega_\ell (n-N)}$ for any bin, this can be further simplified to a form that puts the delay element prior to the multiplier:

$$\xcancel{X_n(\omega_\ell) = X_{n-1}(\omega_\ell) + (x[n] - x[n-N])e^{-j\omega_\ell n}}$$

$$\underline{\hat{X}_n(\omega_\ell) = \hat{X}_{n-1}(\omega_\ell) + (x[n] - x[n-N])e^{-j\omega_\ell n}}$$

This is the form shown in Figure 10B. This structure has several advantages. First only a real delay element is needed. Second, the word width of the delay element is that of the ADC data. Third, in one embodiment, the delay element can be shared for all bins.

Please amend the first paragraph beginning on page 40 as follows:

Figure 17B is a block diagram of a CORDIC processing element 1710 that implements the Type-2 transform shown in Figure 10B. In the processing element 1710, the input b_1 is provided to an input of a register 1701, and the input b_2 is provided to an input of a register ~~1702~~ **1701**. An output of the register 1701 is provided to a first data input, x_k, of the rotation block 1703 and to the output b_1'. An output of the register 1702 is provided to a second input, y_k, of the rotation block 1703 and to the output b_2'. The input θ_k is provided to a rotation input of a CORDIC rotation block 1703. The rotation block 1703 provides the first data output, x_k', and the second data output, y_k'.

Please amend the last paragraph beginning on page 40 as follows:

Figure 19 is a block diagram of a basis function generator 1900 that uses a CORDIC algorithm for sine and cosine generation. In the generator 1900, a frequency/phase control value α is provided to a first input of an adder 1801. An output of the adder 1801 is a signal θ that is provided to an input of a register 1802 and to θ inputs of CORDIC rotation blocks 1901-1904. An output of the register 1802 is provided to a second input of the adder 1801. The CORDIC rotation block 1902 is provided with inputs K and 0 respectively. Data outputs from the CORDIC rotation block 1901 are provided to respective data inputs of the CORDIC ration block 1902. Data outputs from the CORDIC rotation block ~~1902~~ **1901** are provided to respective data inputs of the CORDIC ration block 1903. Similarly, data outputs from the preceding CORDIC rotation block are provided to each successive CORDIC rotation block. Data outputs from the CORDIC rotation block 1904 are sin(θ) and cos(θ) respectively.